Session 3

Hbase, Hive, Pig
Hadoop workshop

Dr. Péter Molnár

Saeid Motevali

Institute for Insight
J. Mack Robinson College of Business
Georgia State University

Background:

- How we get into big data
- Distributed memory
- Shared memory
- Package size (coin example)
- SuperCube

Area effected by parallel processing:

- Since having big data need to process in parallel
- Datamining such as tweets analysists
- Image processing training set
- Algorithm
- Networking

Map Reduce Optimization:

- Before for example just English word.
- On load change the compression ratio, how big the size.
- Map Done by code, Don’t want to use high complex.
- Shuffle highly complex, if you are good try it.
- Reduce reducing process.
- After just give the first result fast, and use the data again for further processing.

Optimization Before Running a job:

- File size. Use the right size.
- Compression. How much can compress and how it does effect the processing.
- Encryption. Encrypting and decrypting takes time

Physical Map Reduce:

- Verify your cluster configuration, and document the reason if not using **default**.
- Unused resources
- Overstress resources (can’t fit in memory and goes to desk)
- Collaboration of local and web data storage

Reducer Optimization:

Subdividing tasks	prevent over flow of memory
Debugging	provided on the nodes that has been used unlike mapper
Spill ration	define how much it goes to disk

What is Pig?

ETL library for Hadoop.

Extract Transform Load

Generate MapReduce

Developed at Yahoo

Example:

- Transform the data: By dividing sentence and collecting word. Classic word count for blog.
- Clean and filtering the data: Such as data for sensor that needs to be clean.
- Process the data: For example data for specific location that you may need.

How Does Pig Works:

Load <file>

Filter, Join, Group By, Foreach, Generate <values>

Dump <to screen for testing>

Store <new file>

Pig data:

- Field: a piece of data
- Tuple: a set of field
- Bag: a collection of tuples
- Pig is complete relation database.

Pig Concepts:

- Filter <set> By <value> = <number>

  Filter A by quantity > 2000;

  Similar to where in relational database

- Supported operations :
  - Logical: NOT, AND, OR
  - Relational: < , > , == , != , >= , <=

Pig Function:

- It is quite powerful and rich, it is worth digging into it.
  - General: AVG, MAX, TOKENIZE
  - Relational: FILTER, MAPREDUCE. can call MAPREDUCE inside a pig script.
  - String: UPERCASE, LOWERCASE
  - Math: ABS, LOG, ROUND
- Write your own function (Write, Register, Test the function in JAVA or PYTHON)

Run Pig:

- Run from Hadoop or pig shell
- Use as embedded within the java code

We should think about mapper and reducer in our code.

What is Hive?

- SQL-like query language that generates MapReduce Code.
- Hive use H-SQL (Hibernate Query Language)
- Developed at Facebook
- Batch, not interactive. means takes time to come up with result.
- It is open source.

NoSql:

- Object oriented
- Beyond the relational database
- Horizontal Scaling, building out instead of up
- mapping

What is HBase?

- Wide-column NoSQL database.
- Use CREATE TABLE over HDFS data.
- It is very different from relational database
- It is distributed, multidimensional sorted map.

Using Hive With Hadoop:

- Hive library are integrated with Hbase.
- Hive libraries include the HQL language.

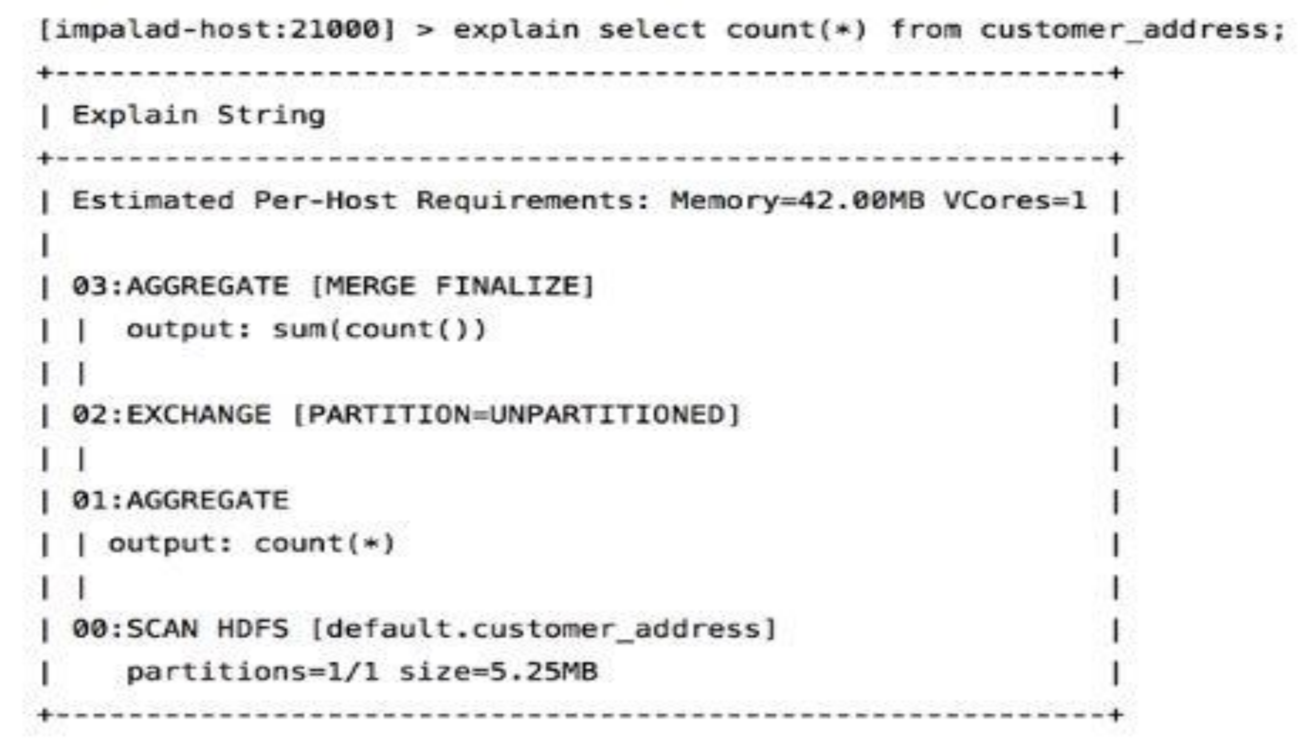
Why Use Hive?

- You are an analyst and you know SQL.
- You want to ask analytical question.
- You work with excel.
- Hive is batch, not interactive. It does produce MapReduce. So, takes time.
- You don't want to do word count by Hive. Pig works better in that manner.

For Hive Query Optimization:

- Partitioning or sampling using subset.
- Cost-based optimization (CBO) by looking at execution method and locating bottleneck.

HQL Query Plan:



```
[impalad-host:21000] > explain select count(*) from customer_address;
+----------------------------------------------------------+
| Explain String                                           |
+----------------------------------------------------------+
| Estimated Per-Host Requirements: Memory=42.00MB VCores=1 |
|                                                          |
| 03:AGGREGATE [MERGE FINALIZE]                            |
| |  output: sum(count())                                  |
| |                                                        |
| 02:EXCHANGE [PARTITION=UNPARTITIONED]                    |
| |                                                        |
| 01:AGGREGATE                                             |
| | output: count(*)                                       |
| |                                                        |
| 00:SCAN HDFS [default.customer_address]                  |
|    partitions=1/1 size=5.25MB                            |
+----------------------------------------------------------+
```

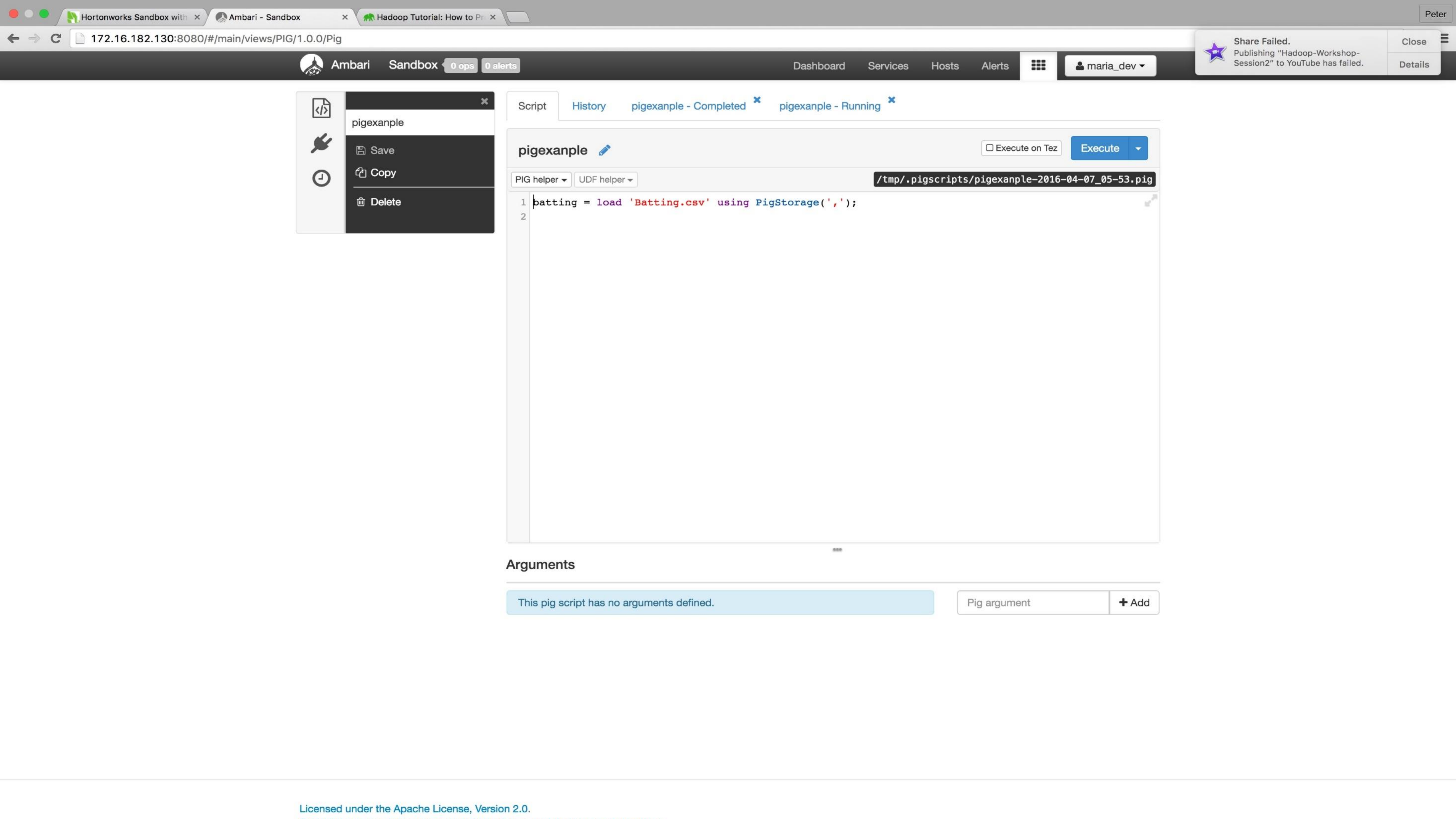
Works To Do:

- Fire up your vmware or your virtualbox.
- If you have no memory open cloudera.
- Open the browser and type the address.
- Use maria_dev as your username and password
- Go to this link and go through the steps for Pig:

http://hortonworks.com/hadoop-tutorial/how-to-process-data-with-apache-pig/

Go to this link and go through the steps for Hive:

http://hortonworks.com/hadoop-tutorial/how-to-process-data-with-apache-hive/



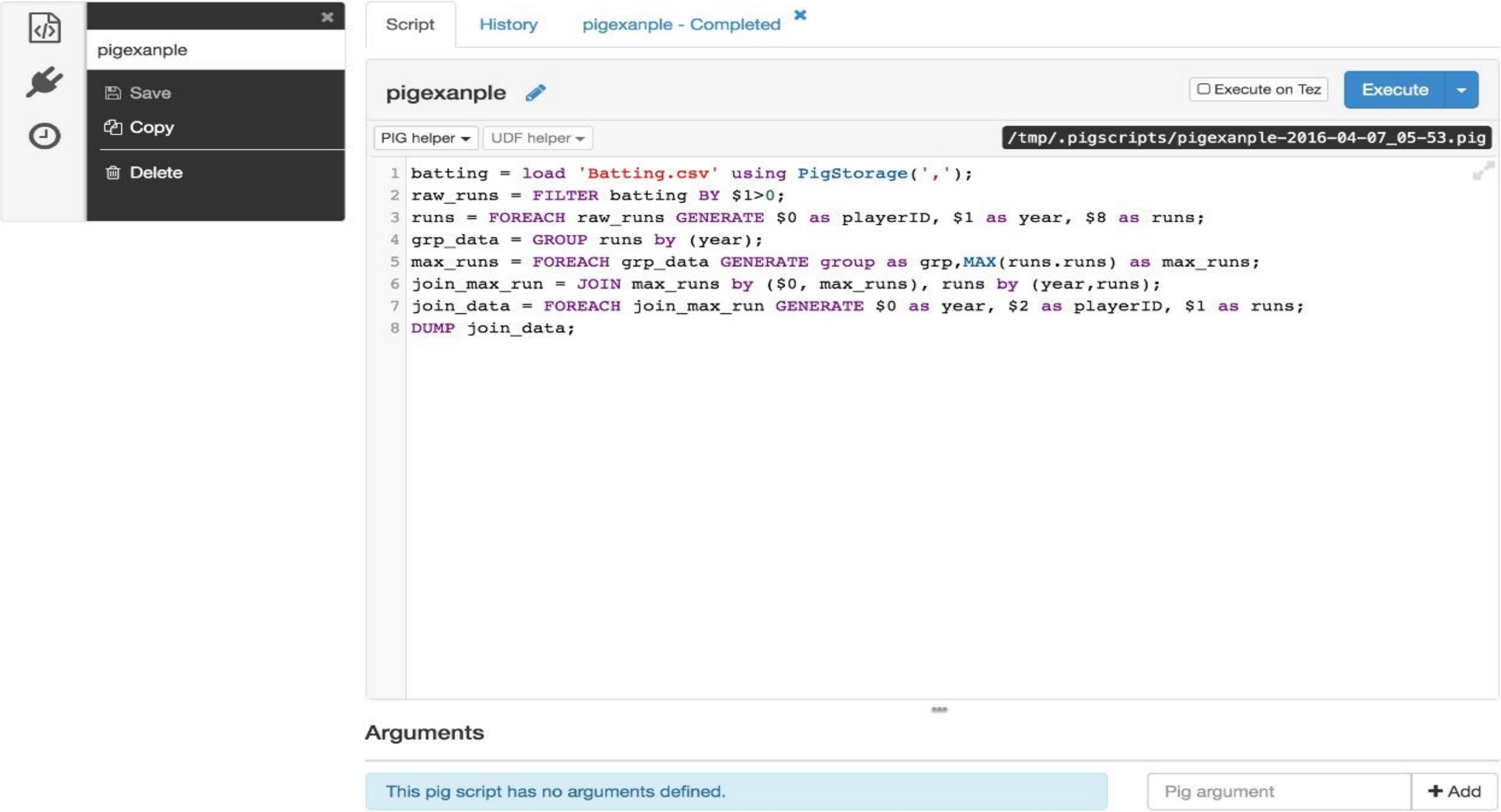
Share Failed.
Publishing "Hadoop-Workshop-Session2" to YouTube has failed.

/tmp/.pigscripts/pigexanple-2016-04-07_05-53.pig

```
batting = load 'Batting.csv' using PigStorage(',');
```

This pig script has no arguments defined.

Licensed under the Apache License, Version 2.0.

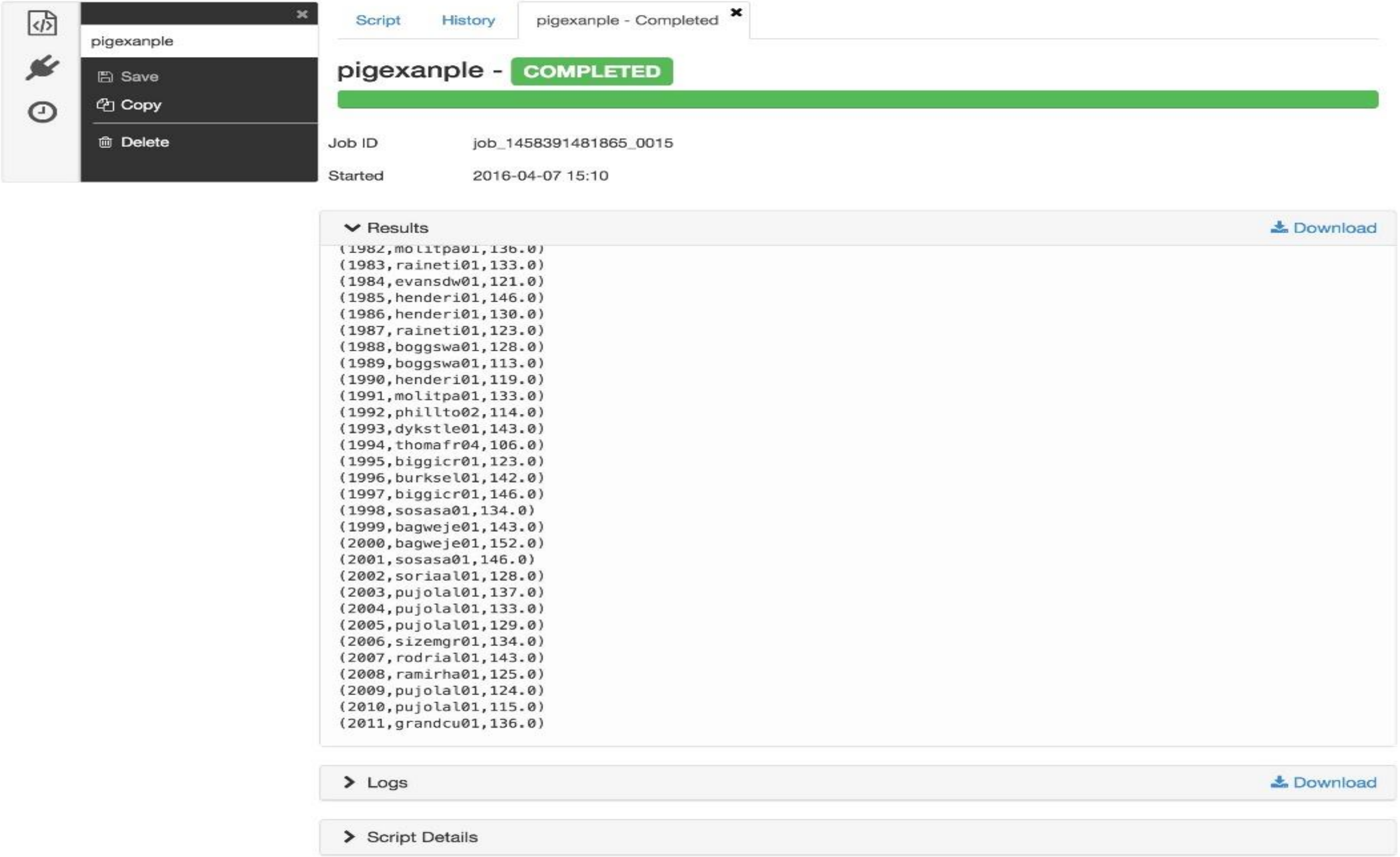


pigexanple

/tmp/.pigscripts/pigexanple-2016-04-07_05-53.pig

```
batting = load 'Batting.csv' using PigStorage(',');
raw_runs = FILTER batting BY $1>0;
runs = FOREACH raw_runs GENERATE $0 as playerID, $1 as year, $8 as runs;
grp_data = GROUP runs by (year);
max_runs = FOREACH grp_data GENERATE group as grp,MAX(runs.runs) as max_runs;
join_max_run = JOIN max_runs by ($0, max_runs), runs by (year,runs);
join_data = FOREACH join_max_run GENERATE $0 as year, $2 as playerID, $1 as runs;
DUMP join_data;
```

This pig script has no arguments defined.

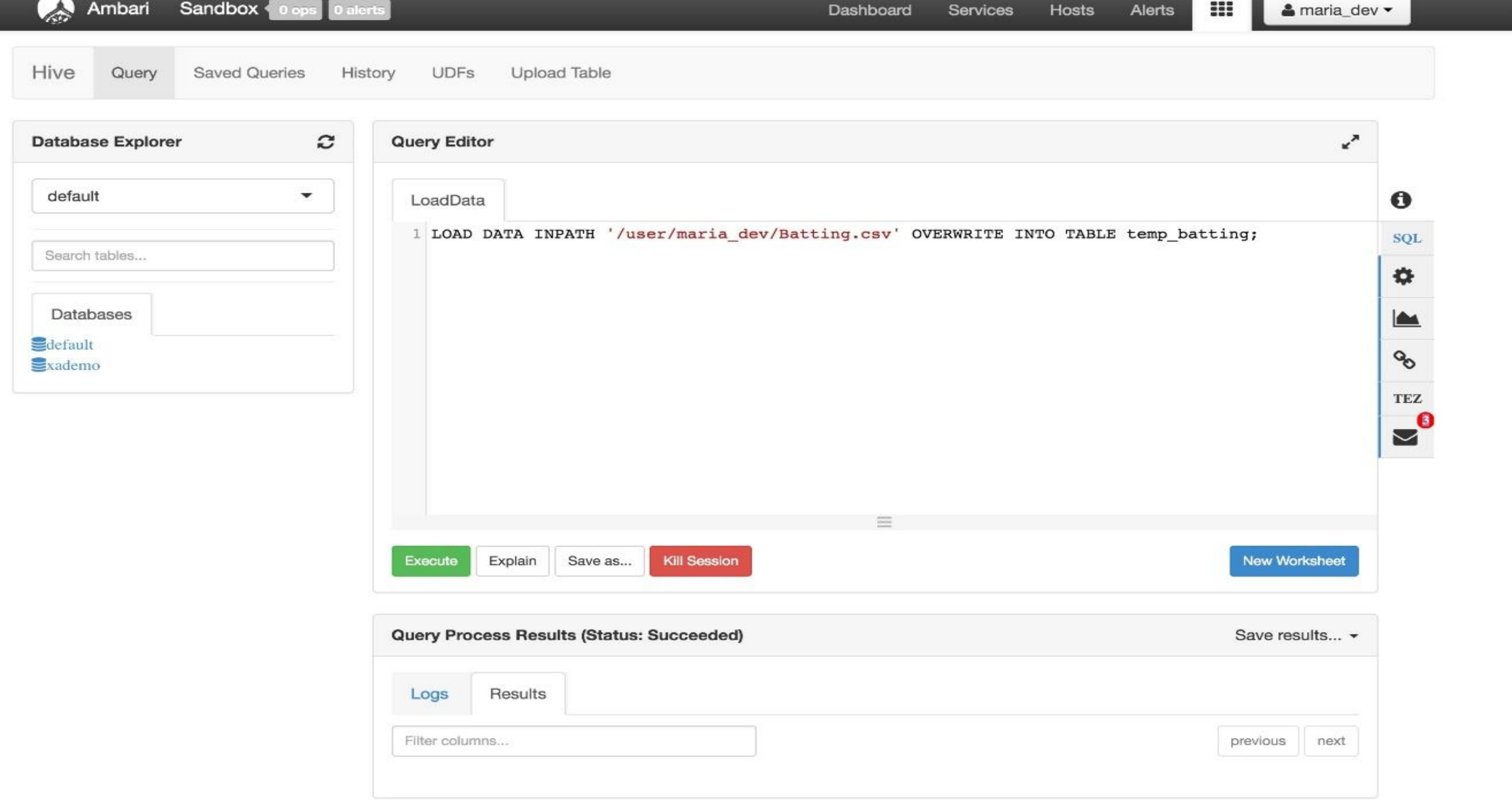


pigexanple - COMPLETED

Job ID job_1458391481865_0015

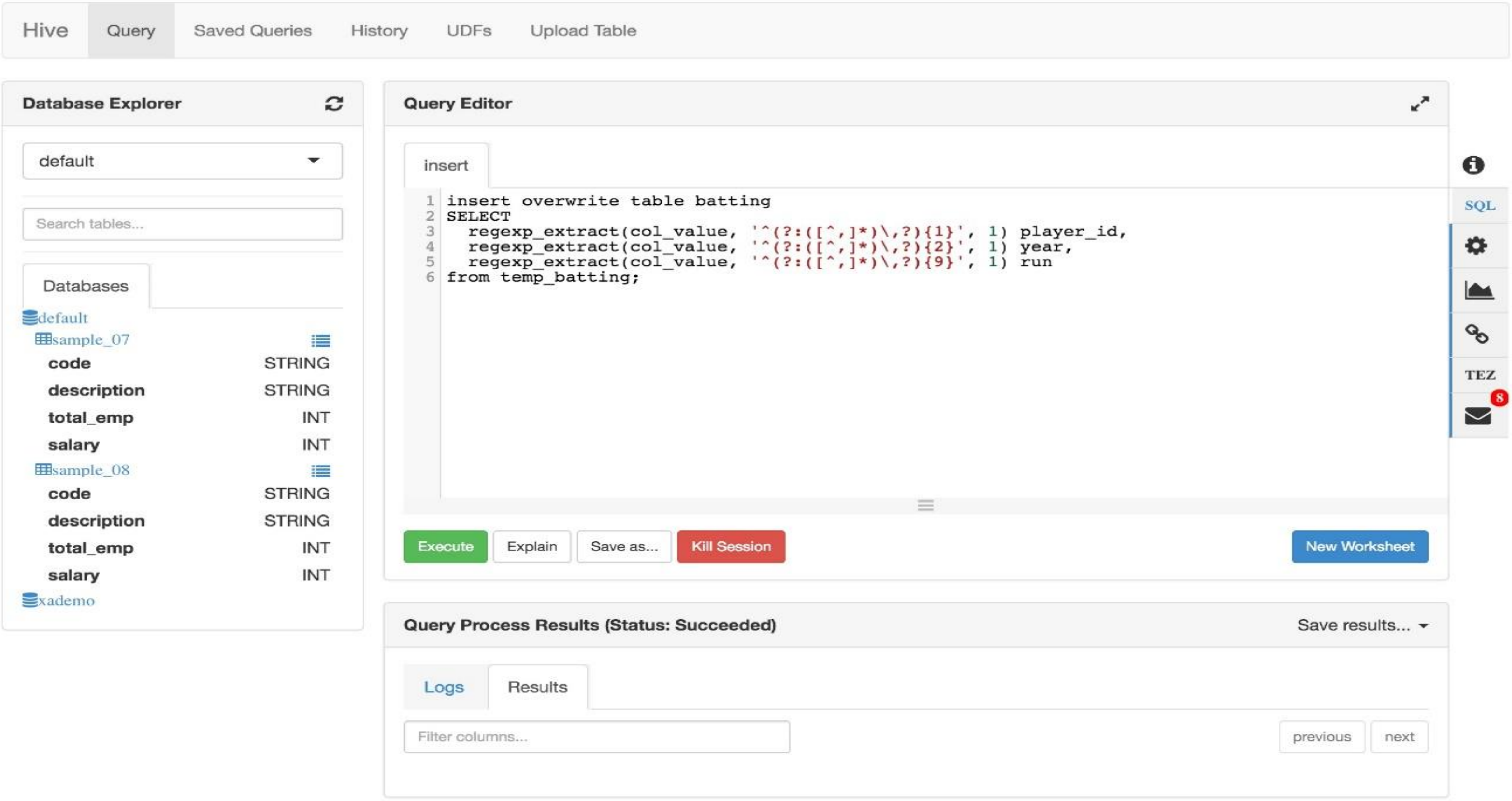
Started 2016-04-07 15:10

```
(1982,molitpa01,136.0)
(1983,raineti01,133.0)
(1984,evansdw01,121.0)
(1985,henderi01,146.0)
(1986,henderi01,130.0)
(1987,raineti01,123.0)
(1988,boggswa01,128.0)
(1989,boggswa01,113.0)
(1990,henderi01,119.0)
(1991,molitpa01,133.0)
(1992,phillto02,114.0)
(1993,dykstle01,143.0)
(1994,thomafr04,106.0)
(1995,biggicr01,123.0)
(1996,burksel01,142.0)
(1997,biggicr01,146.0)
(1998,sosasa01,134.0)
(1999,bagweje01,143.0)
(2000,bagweje01,152.0)
(2001,sosasa01,146.0)
(2002,soriaal01,128.0)
(2003,pujolal01,137.0)
(2004,pujolal01,133.0)
(2005,pujolal01,129.0)
(2006,sizemgr01,134.0)
(2007,rodrial01,143.0)
(2008,ramirha01,125.0)
(2009,pujolal01,124.0)
(2010,pujolal01,115.0)
(2011,grandcu01,136.0)
```



```
LOAD DATA INPATH '/user/maria_dev/Batting.csv' OVERWRITE INTO TABLE temp_batting;
```

Query Process Results (Status: Succeeded)



default

sample_07

code	STRING
description	STRING
total_emp	INT
salary	INT

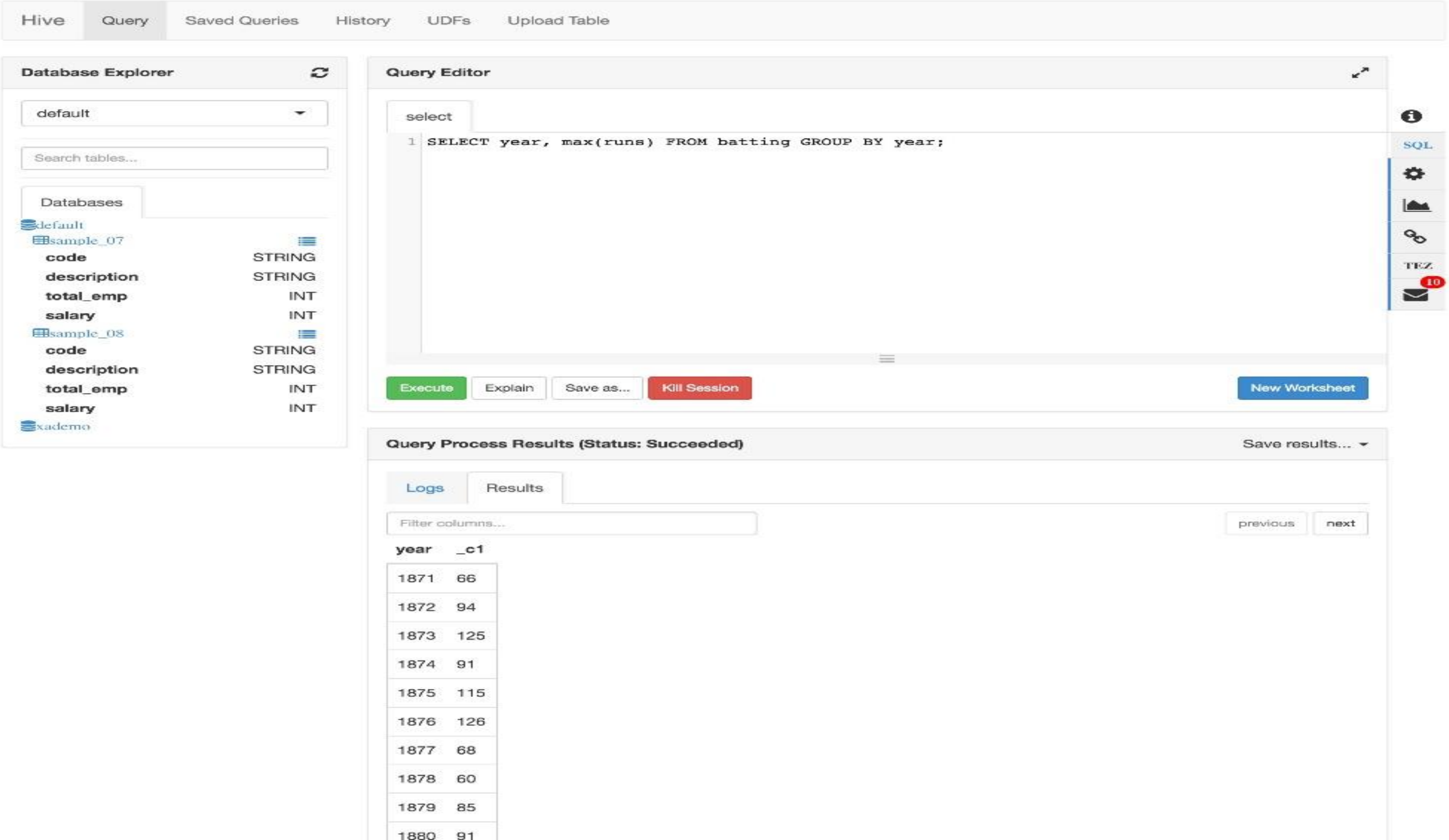
sample_08

code	STRING
description	STRING
total_emp	INT
salary	INT

xademo

```
insert overwrite table batting
SELECT
  regexp_extract(col_value, '^(?:([^,]*)\,?){1}', 1) player_id,
  regexp_extract(col_value, '^(?:([^,]*)\,?){2}', 1) year,
  regexp_extract(col_value, '^(?:([^,]*)\,?){9}', 1) run
from temp_batting;
```

Query Process Results (Status: Succeeded)



default

default

sample_07

code	STRING
description	STRING
total_emp	INT
salary	INT

sample_08

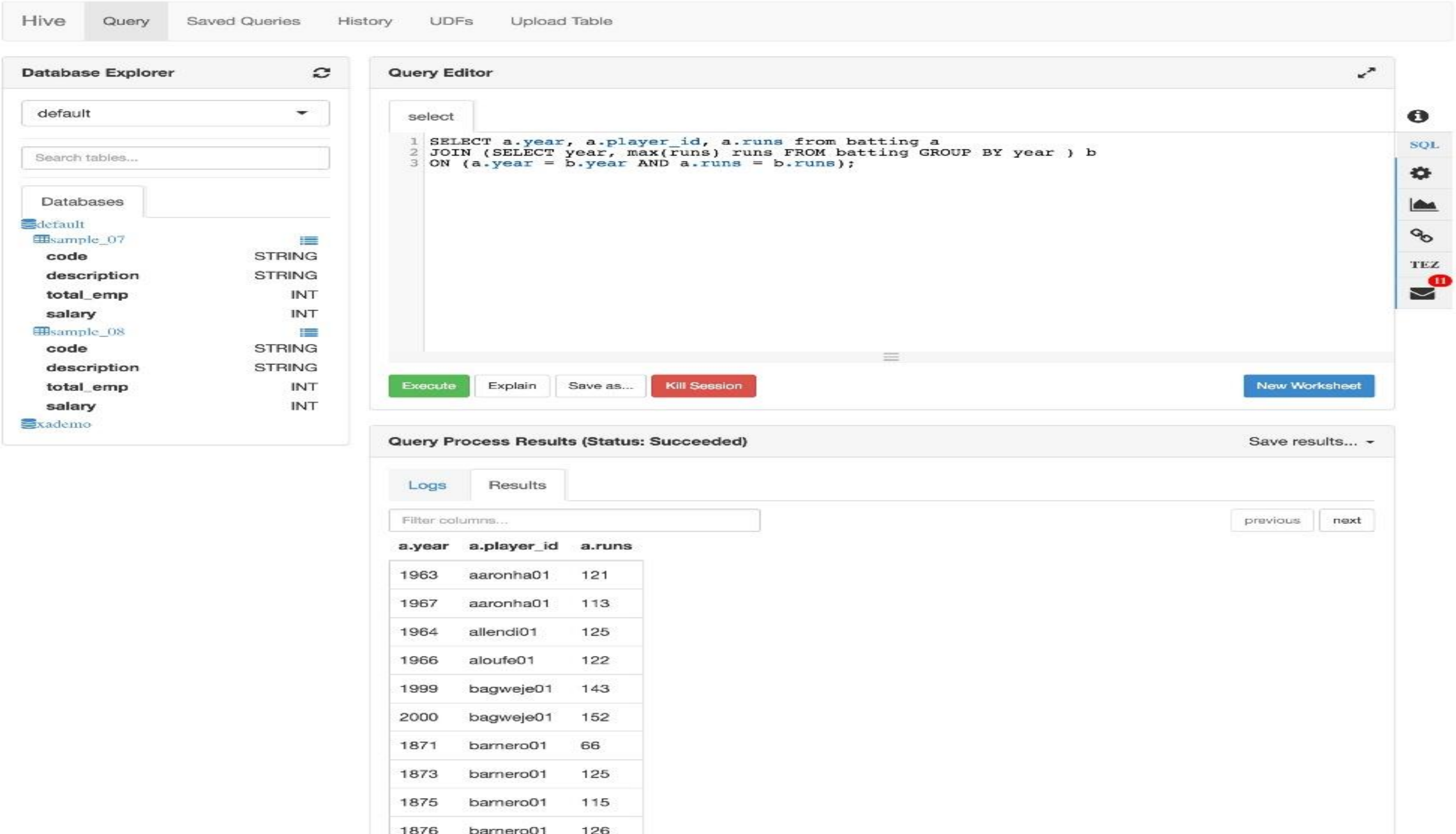
code	STRING
description	STRING
total_emp	INT
salary	INT

xademo

```
SELECT year, max(runs) FROM batting GROUP BY year;
```

Query Process Results (Status: Succeeded)

year	_c1
1871	66
1872	94
1873	125
1874	91
1875	115
1876	126
1877	68
1878	60
1879	85
1880	91



default
sample_07

code	STRING
description	STRING
total_emp	INT
salary	INT

sample_08

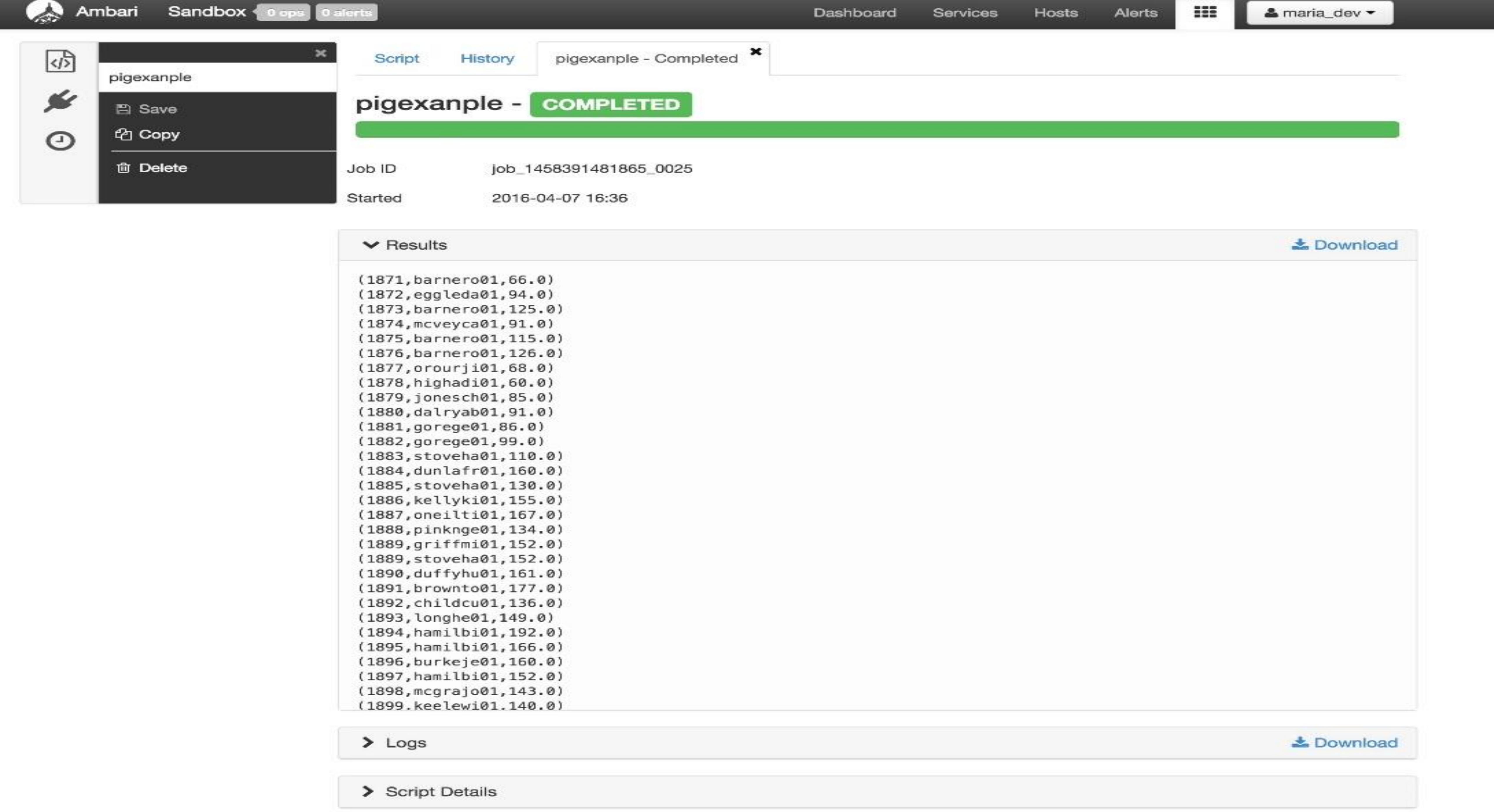
code	STRING
description	STRING
total_emp	INT
salary	INT

xademo

```
SELECT a.year, a.player_id, a.runs from batting a
JOIN (SELECT year, max(runs) runs FROM batting GROUP BY year ) b
ON (a.year = b.year AND a.runs = b.runs);
```

Query Process Results (Status: Succeeded)

a.year	a.player_id	a.runs
1963	aaronha01	121
1967	aaronha01	113
1964	allendi01	125
1966	aloufe01	122
1999	bagweje01	143
2000	bagweje01	152
1871	barnero01	66
1873	barnero01	125
1875	barnero01	115
1876	barnero01	126



pigexanple - COMPLETED

Job ID job_1458391481865_0025

Started 2016-04-07 16:36

```
(1871,barnero01,66.0)
(1872,eggleda01,94.0)
(1873,barnero01,125.0)
(1874,mcveyca01,91.0)
(1875,barnero01,115.0)
(1876,barnero01,126.0)
(1877,orourji01,68.0)
(1878,highadi01,60.0)
(1879,jonesch01,85.0)
(1880,dalryab01,91.0)
(1881,gorege01,86.0)
(1882,gorege01,99.0)
(1883,stoveha01,110.0)
(1884,dunlafr01,160.0)
(1885,stoveha01,130.0)
(1886,kellyki01,155.0)
(1887,oneilti01,167.0)
(1888,pinknge01,134.0)
(1889,griffmi01,152.0)
(1889,stoveha01,152.0)
(1890,duffyhu01,161.0)
(1891,brownto01,177.0)
(1892,childcu01,136.0)
(1893,longhe01,149.0)
(1894,hamilbi01,192.0)
(1895,hamilbi01,166.0)
(1896,burkeje01,160.0)
(1897,hamilbi01,152.0)
(1898,mcgrajo01,143.0)
(1899,keelewi01,140.0)
```